

Chapter 2

Background: Asic Design Methodologies

Integrated Circuit (IC) is a set of electronic devices on a die, made from semiconductor material, normally silicon wafer, each of them holding hundreds of dies. Circuits are built up in many overlapping layers of materials like polysilicon, aluminum, and silicon dioxide, each defined by photolithography. Patterns must be etched into the material to create transistors and interconnections on the surface of the wafer. Etching creates microscopic patterns on the wafer's surface, and is the true magic of IC technology because it delineates fine geometries using an optical process, forming very small features.

Application Specific Integrated Circuit (ASIC) is a subset of Integrated Circuits (ICs) that are designed and customized for a particular function specified by the owner of the ASIC for its sole use, rather than intended for general-purpose (microprocessors, memories, and etc.) use. ASIC can be software programmable to perform a wide variety of different tasks, while being a digital circuit built up with successive mask layers to meet specifications set by a specific function.

IC design (organic/inorganic) is an interactive process with given trade-offs between die area (highly related to cost), complexity, performance (speed, power, etc.), yield, energy efficiency and design time. Considering the trend toward steadily miniaturization and enhanced complexity as predicted by Moore's law, the target for the development of ASIC is to optimize the involved fabrication processes regarding to the above-mentioned criteria what leads also to technology specialization.

There are several types of existing ASIC design methodologies, either of which is prone to some cost functions. In this chapter, an overview of ASIC design methodologies with their advantages and disadvantages is presented. Figure 2.1 shows the evolution of microelectronics circuit design methodologies from technology-dependent full-custom and cell-based design, to technology-independent re-usable design, during its evolution years. It also indicated the complexity of final prototype in respect to the transistor counts [1].

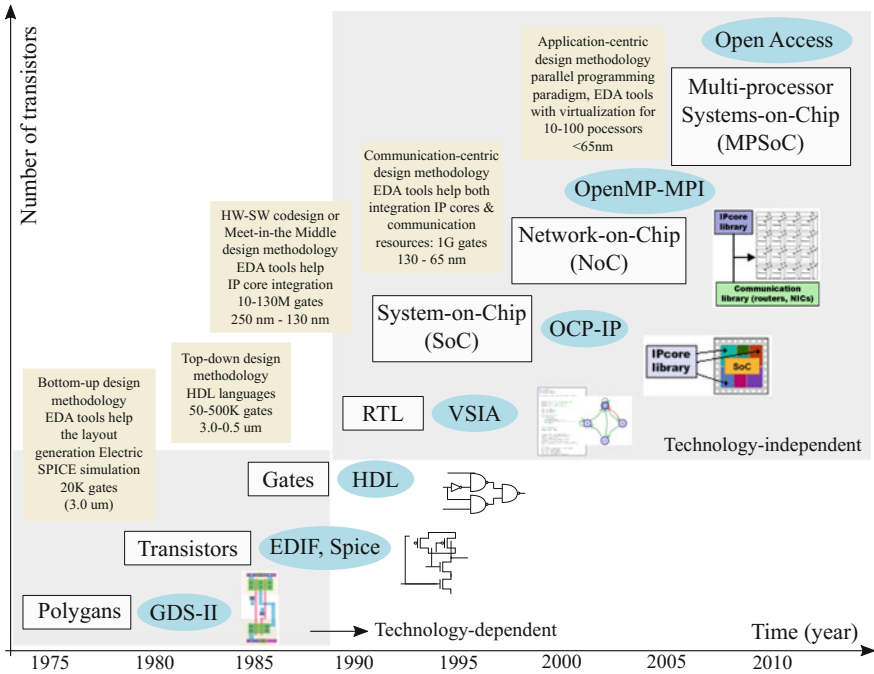


Fig. 2.1 Microelectronic evolution: from geometries to systems

2.1 Full-Custom

The oldest and most traditional circuit design technique is termed full-custom, in which a designer sits in front of a graphics display running an interactive editor and pieces designs together at the geometry level one rectangle at a time. This work is sometimes called polygon pushing and the variation of custom mask design is called symbolic layout. In this methodology, the layout of each transistor and the interconnections between them is specified.

Full-custom design is the lowest abstraction level from design point of view, which uses the basic set of information (electrical parameters and models, geometrical design rules, and technology layers) to design, simulate and draw circuit. Rather than dealing with rectangles and polygons on various mask levels, the primitives such as transistors, contacts, wires, and ports (points of connection) can be used in custom design.

Custom design is worthwhile pedagogically because it completes the link between technology and design: from technology layout transistors, cells, and systems. Its main advantage is maximized performance and minimized area of the chip, but it is extremely labor-intensive and risky to implement and several mask sets are required

in order to transfer the circuit designs onto the wafer. Due to area and performance optimization in custom design, the scope of the methodology covers the key design domains of analog and RF, custom digital, and cell libraries for standard cell libraries.

2.2 Cell-Based

Cell-based design methodology is a semi-custom method of designing ASICs with digital logic cells. The basic building blocks of a chip are pre-designed and pre-characterized (standard cells) or pre-fabricated (gate arrays). This is an example of design abstraction, where a low-level integration is encapsulated into abstract logic representation. The cells are placed in appropriate positions, then their interconnections are routed. Foundries and library vendors supply cells with a wide range of functionalities such as:

1. Small-scale integration (SSI) digital logic (NAND, NOR, XOR, AOI, OAI, inverters, buffers, registers) and analog cells (current mirrors, opamps, ADC & DAC, and etc.)
2. Memories (RAM, ROM, CAM, register files), usually built out of compilers for structures of elementary cells.
3. System level modules such as processors, protocol processors, serial interfaces, and bus interfaces, known as IPs and built out of two previous elements.
4. Possibility of mixed-signal and RF modules

A typical standard cell library contains two main components:

1. Library database that consists of a number of views including layout, schematic, symbol, abstract, and other simulation views (behavioral, electrical, and etc.). They can be provided through various information means including the Cadence LEF format, and Synopsys Milkway format. LEF file contains reduced information about the cell layouts, sufficient to run automated “place and route” tools.

2. Timing abstract, which is typically in liberty format, to provide functional definitions, timing, power, and noise information for each cell.

Standard cells are fixed-height, variable-width with power and ground routed respectively at the top and bottom of the cells. This allows the cells to be abutted end to end and to have the supply rails connect, easing the process of automated digital layout. The cells are typically optimized full-custom, in order to minimize the delays and area. This design methodology can deliver small, fast, and low-power chips to produce the custom mask set. Therefore, it is only economical for high volume parts or when the performance commands a lucrative sales price. As compared to full-custom design that required even higher costs (and thus usually volume), it offers much higher productivity because it uses pre-designed cells with layouts.

2.3 Gate Arrays and Sea-of-Gates

Designers typically strive to keep the NRE cost as low as possible. The Gate Array (GA) design alternative is a semi-custom prefabricated silicon chip, containing a common base array of transistors or logic gates with no function. These logic blocks are placed at regular predefined positions, surrounded by interconnect resources, and are kept unconnected. Later on, creation of a circuit with specific functionality is accomplished by altering the metallization (metal and via masks) layers that is placed on top of the arrays.

The key feature in GA design methodology is that the master slices are usually prefabricated and stockpiled in large quantities regardless of the customer orders. The design and fabrication according to end-user application is realized by using a reduced metallization step(s), which alleviates the often prohibitive time and expense of multiple mask set designs in cell-based and full-custom design. In this way, the costs of the shared masks are spread over all applications and only the cost of the final customization mask is additional.

A particular subclass of GAs is known as a Sea-of-Gates (SoG) chip (channel-less gate array), in which rows of nMOS and pMOS transistors are arrayed in the chip. Each logic row consists of an n-row and p-row. Personalization of SoG structure commences at contact and metal1 masks, and can continue up for all metal layers available in the process. Figure 2.2 depicts a sea of gates manufacturing steps.

There are two basic limitations of gate arrays: firstly, the designer or user cannot customize the chip in-the-field and the customization is performed during chip fabrication by specifying the metal interconnect. Secondly, the gate arrays are One-Time Programmable (OTP) devices, which means once the chip is customized to its function, it is not possible to erase that functionality and re-configure it.

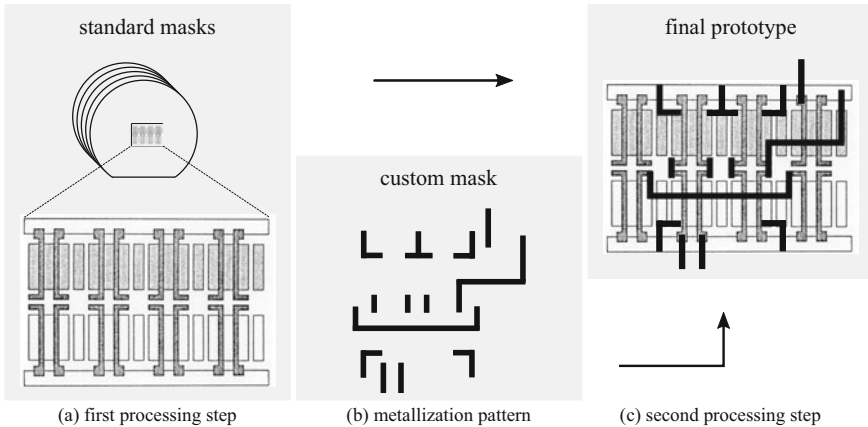


Fig. 2.2 Manufacturing steps of a sea of gates chip

2.4 Field-Programmable Devices

Field-Programmable Device (FPD), also known as field-configurable device, or Programmable Logic Device (PLD), refers to any type integrated circuit used for implementing digital hardware, where the chip can be configured by the end user in-the-field to realize different designs. Programming of such a device often involves placing the chip into a special programming or configured “in-system” using standard means (USB, JTAG, etc.). The most compelling advantages of FPDs are instant manufacturing turnaround, low start-up costs, low financial risk (since programming is done by the end user) and ease of design.

Field-Programmable Gate Array (FPGA) is an FPD, which consists of an array of logic cells surrounded by configurable routing resources similar to gate arrays, but the configuration is generally specified using a Hardware Description Language (HDL) or composing IP blocks. They use the high circuit densities in modern processes featuring a general structure that allows very high logic capacity. There are two basic versions of FPGAs in respect to programming method.

The first uses a special process option such as a fuse or antifuse to permanently program interconnect and personalize logic, addressed as OTP. As an example, devices manufactured by Actel embed an array of logic modules within an interconnect matrix that is formed on the top metal layers [2]. These successive routing channels run vertically or horizontally, and an antifuse OTP contacts are placed at the intersection of routing traces. Figure 2.3 shows the antifuse that consists of three sandwiched layers (conductor-insulator-conductor) and is located between two interconnect wires. It normally has high resistance (effectively open circuit), and by applying a special programming voltage across the contact, the resistance permanently drops to a few ohms. Actel’s antifuses uses Poly-Si and n^+ diffusion as conductor and ONO [2], while other antifuses rely on metal for conductors, with amorphous silicon as the middle layer [3].

The advantage of this type of routing is that the size of the programmable interconnect is tiny enabling higher overall density. The disadvantage is that the interconnect is not re-programmable, so once a chip is programmed, its function is fixed to the extent that the interconnect has been personalized [4].

The second type of FPGAs uses Static Random Access Memory (SRAM) or flash memory to configure routing and logic functions. The key advantage that this type

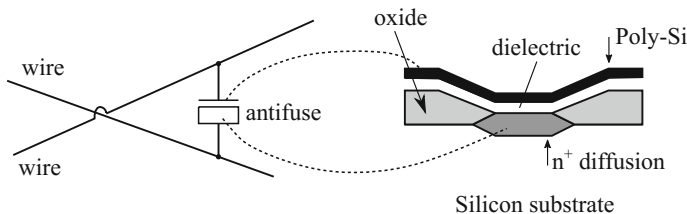


Fig. 2.3 Actel antifuse structure

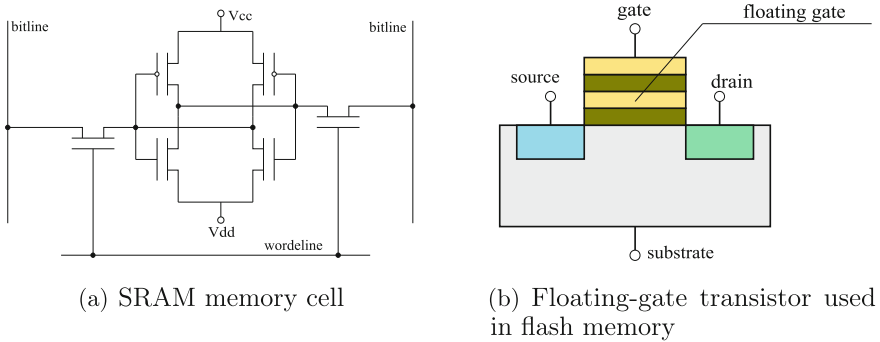


Fig. 2.4 Re-programmable configuration techniques

of configuration offers is the re-programmability feature, which allows the end user to reuse the chip for different applications.

SRAM-based FPGAs store logic cells configuration data in the static memory (organized as an array of latches). SRAM is volatile and cannot keep data without power supply, therefore, the FPGA must be configured upon start. While it may seem inefficient (in terms of area, speed and power) to use a RAM cell to perform logic, specially designed single-data line RAMs are small and fast in current processes, and resources such as the routing tend to dominate modern designs from a density and speed viewpoint. Currently, Xilinx Virtex [5], and Spartan families [6], and Altera Stratix [7] and Cyclone [8] devices use SRAMs for programming. Figure 2.4a shows an SRAM memory cell.

Flash-based FPGAs use flash as a primary resource for configuration storage. Flash memories require more complex process than SRAM. This technology has an advantage of being less power consumptive, and more tolerant to radiation effects. Furthermore, this type of FPGAs has a non-volatile memory cell to hold the configuration pattern right on the chip, and even if power is removed the contents of the flash cells stay intact. FPGA families, such as Actel LGLOO and ProASIC3 use this configuration technique. Figure 2.4b shows a floating-gate transistor used in flash memory. Traditionally, due to challenges of shrinking the flash memory cell, flash-based FPGAs used to suffer from lower density and performance than SRAMs. However, advances in process technology now allow the FPGA designers to shrink the flash configuration cells and integrate them into advanced logic processes, enabling high-performance flash-based FPGAs to deliver features and functions comparable to or even better than SRAM-based FPGAs.

2.5 Altera HardCopy

Traditionally, ASICs have been the ideal solution for system architects designing high-volume, high-performance applications. However, designing complex ASICs requires expensive design tools, results in high development costs, and involves significant overall risk when bringing products to market in a timely manner. In contrast, FPGAs offer design flexibility and automation by using software to estimate performance and power consumption, and maximize system throughput. Moreover, FPGAs reduce the risk, time-to-market, and cost. However, comparing with high-density ASIC, they consume more power with lower performance. They are also not very common and cost-efficient for volume production.

As a mid-point solution, Altera [9] introduced new devices known as Altera's HardCopy devices, that offer system solution by providing flexibility of FPGAs at the low power and cost of ASICs for volume production. One example is Altera's HardCopy Stratix™ series [10] that preserve their Stratix FPGA counterpart's architecture [7], but the configuration and programmable routing resources are removed and replaced with direct metal interconnects (same concept as metallization step in SoGs). This results in considerable die size reduction and ensuing cost savings. The user can obtain an average of 50% higher performance and up to 40% lower power consumption than can be achieved in the corresponding Stratix FPGAs. However, once the device is manufactured, the functionality of the device is fixed and no re-programming is possible.

HardCopy Stratix devices consist of base arrays that are common to all designs for a particular device density and design-specific customization is done within the top metal layer(s). The base arrays use an area-efficient Sea-Of-Logic-Elements (SOLE) core and extend the flexibility of high-density Stratix FPGAs to a cost-effective, high volume production solution. With a seamless migration process, functionality-verified Stratix FPGA designs can be migrated to fixed-function HardCopy Stratix devices with minimal risk and guaranteed first-time success. Another family of Altera's HardCopy devices is APEX™ [11] manufactured using an 0.18 μ m CMOS six-layer-metal process technology, which enables high-density APEX 20KE device technology [12] to be used in high-volume applications.

To ensure the device functionality and performance, designers should thoroughly test the original FPGA-based design for satisfactory results before committing the design for migration to a HardCopy device. For that, Altera has developed a "design once" flow for both HardCopy ASICs and their prototyping FPGAs. At the front end, before design hand-off, user can have one design, one IP set, one methodology, and one tool to create two device implementations. First, the user can get his system ready for production with an FPGA prototype, and guarantee a functional equivalent and pin-/footprint-compatible HardCopy ASIC as a drop-in replacement for volume production (Fig. 2.5).

Once the system is ready, one can perform market testing and initial low-volume production with FPGA devices. Following that, if the client requires low-power, high-performance, volume production of the same system, then HardCopy ASICs

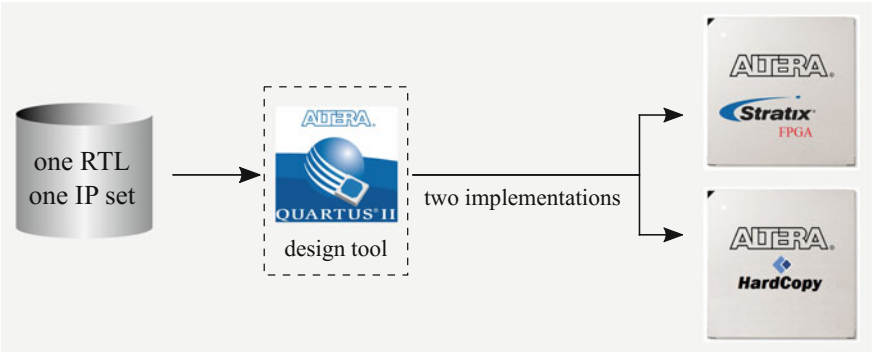


Fig. 2.5 One design, one RTL, one IP set, two implementations

can be employed to obtain a fully-tested and production-quality sample in less than two weeks and the large-volume production in twelve weeks.

Software, such as, Quartus II provides a complete set of inexpensive and easy-to-use tools for designing HardCopy devices in a manner similar to the traditional ASIC design flow. Combination of FPGAs for prototyping and design verification, HardCopy devices for high-volume production, and the design software provide a comprehensive and powerful alternative to conventional ASIC.

2.6 Summary

Several ICs, as well as their pros and cons were explained in this chapter. The main factor in choosing a class of IC is the quantity of parts you will need in mass production, because selling a larger volume of parts drives the need for a lower unit cost and provides profits to pay back a large NRE. High volume ICs need a low unit price but can tolerate a high NRE, so full custom ICs are the best fit. Low volume ICs can tolerate a high unit cost but need a low NRE, so FPGAs are the best choice. Semi-custom ASICs (standard cells and gate arrays) provide a balance of reasonable unit cost and moderate NRE, so they fit in the middle ground where production volumes are moderate. Table 2.1 indicates the proper design methodology in respect to annual production volume (Source: [13]).

Table 2.1 Selecting an integrated circuit class

Annual production	Critical factors	IC class
High	Low unit cost	Full-custom
Medium	Reasonable unit cost and NRE	Semi-custom (SC and GA)
Low	Low NRE and quick time to develop	FPGA

Table 2.2 Characteristics of integrated circuit classes

Characteristic	Full-custom IC	Semi-custom IC	FPGA
Masks customized	All	A few	None
NRE	\$300,000 to \$millions	\$50,000 to \$150,000	Less than \$50,000
Die size	Small	Medium	Large
Time to develop	2 to 5 years	About 1 year	A week to a few months
Unit cost (10M/year)	Small	Medium	Large
Unit cost (100K/year)	Not feasible	Medium	Large
Unit cost (1K/year)	Not feasible	Not feasible	Large

Another important factor in choosing a class of IC is the time it takes to develop the chip. Development time may be a few months for FPGAs, a year or so for standard cells and gate arrays, and few years for full custom ICs. Some design methodologies take advantage of the best features of all three classes of ICs by prototyping with FPGAs, beginning low volume production with ASICs, and converting the design to full custom if the product is successful in the market and the volumes ramp up to millions. Table 2.2 summarizes the characteristics of classes (Source: [13]).

Due to a lack of circuit components, cell libraries, and an automated design flow, the advantages of a cell-based and array-based semi-custom design methodology are not yet explored when considering digital design for organic electronics. In this thesis, we use the above-mentioned well-established benefits of ASIC industry and map it to Application Specific Printed Electronic Circuits (ASPEC) by adopting array-based design methodology.

References

1. L. Terés, Head of Integrated Circuits and Systems (ICAS), Group at IMB-CNM, Barcelona-Spain
2. E. Hamdy, J. McCollum, S.-O. Chen, S. Chiang, S. Eltoukhy, J. Chang, T. Speers, A. Mohsen, Dielectric based antifuse for logic and memory ICs, in *Electron Devices Meeting, IEDM'88. Technical Digest, International*, (IEEE 1988) pp. 786–789
3. J.a. Birkner, A. Chan, H. Chua, A. Chao, K. Gordon, B. Kleinman, P. Kolze, R. Wong, A very-high-speed field-programmable gate array using metal-to-metal antifuse programmable elements. *Microelectron. J.* 23(7), 561–568 (1992)
4. D.M. Harris, N.H.E. Weste, *CMOS VLSI Design, A Circuit and System Perspective* (Addison-Wesley, USA, 2011)
5. Xilinx, <http://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascaleplus.html>, Xilinx Vitix FPGAs
6. Xilinx, <http://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html>, Xilinx Spartan FPGAs
7. Altera stratix series, <https://www.altera.com/products/fpga/stratix-series.html>, Stratix FPGAs and SOCs

8. Altera, <https://www.altera.com/products/fpga/cyclone-series.html>, Altera Cyclone FPGAs
9. Altera, <https://www.altera.com/>, Altera FPGAs and SOCs
10. Altera stratix series, <https://www.altera.com/support/support-resources/operation-and-testing/dev-hardcopystrx.html>, Stratix FPGAs and SOCs
11. Altera APEX 20K device family, <https://www.altera.com/support/supportresources/operation-and-testing/dev-hardcopyapex.html>, APEX 20k ASIC family
12. Altera APEX 20K device family, <https://www.altera.com/products/general/devices/apex/overview.html>, APEX FPGAs and SOCs
13. E. Rhodes, *ASIC Basics: Black and White Edition* (Lulu.com, USA, 2008)

<http://www.springer.com/978-3-319-72115-6>

Inkjet-Configurable Gate Array

Towards Application Specific Printed Electronic Circuits

Mashayekhi, M.

2018, XV, 209 p. 165 illus., 129 illus. in color.,

Hardcover

ISBN: 978-3-319-72115-6